

Energy-balancing passivity-based control through reinforcement learning

O.R. Sprangers, G.A.D. Lopes, R. Babuška

Delft Center for Systems & Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands

Abstract

Passivity-based control for port-Hamiltonian systems provides an intuitive way of achieving stabilization by rendering a system passive with respect to a desired storage function. However, in most instances the control law has to be calculated by solving a complex partial differential equation (PDE). This paper considers energy-balancing passivity-based control (EB-PBC), which is a form of PBC in which the closed-loop energy is equal to the difference between the stored and supplied energies. We propose a method to parameterize EB-PBC that preserves the systems' PDE matching conditions, does not require the specification of a global desired Hamiltonian, includes performance criteria, and is robust to extra non-linearities such as control input saturation. The parameters of the control law are found using actor-critic reinforcement learning, enabling learning near-optimal control policies satisfying a desired closed-loop energy landscape. The advantages are that near-optimal controllers can be generated using standard energy shaping techniques and that the solutions learned can be interpreted in terms of energy shaping and damping injection, which makes it possible to numerically assess stability using passivity theory. From the reinforcement learning perspective, our proposal allows for the class of port-Hamiltonian systems to be incorporated in the actor-critic framework, speeding up the learning thanks to the resulting parameterization of the policy. The method has been successfully applied to the pendulum swing-up problem in simulations and real-life experiments.

Key words: port-Hamiltonian systems, passivity-based control, energy balancing, reinforcement learning, actor-critic

1 Introduction

Passivity-based control (PBC) [16] is a methodology that achieves the control objective by rendering a system passive with respect to a desired storage function [17]. Different forms of PBC have been successfully applied to design robust controllers [23] for mechanical systems and electrical circuits [17,18]. A key feature of PBC is that it exploits structural properties of the system. In this paper, we are interested in the passivity-based control of systems endowed with a special structure, called port-Hamiltonian (PH) systems. PH systems have been widely used in PBC applications [5,19]. Their geometric structure allows reformulating a PBC problem in terms of solving a set of partial differential equations (PDE's). Much research in the literature concerns solving or simplifying such generally complex PDE's [17].

The drive for passivity-based control of port Hamiltonian systems is grounded in the search for global stabil-

ity, thus strongly relying on models. Other control techniques have been developed when no models are known and performance is important. One such example is reinforcement learning (RL) [21]. RL is a semi-supervised learning control method that can solve optimal (stochastic) control problems for nonlinear systems, without the need for a process model or for explicitly solving complex equations. In RL the controller receives an immediate numerical reward as a function of the process state and possibly control action. The goal is to find an optimal control policy that maximizes the cumulative long-term rewards, which corresponds to maximizing a value function [21]. In this paper, we use actor-critic techniques [11], which are a class of RL methods in which a separate actor and critic are learned. The critic approximates the value function and the actor the policy (control law). Actor-critic reinforcement learning is suitable for problems with continuous state and action spaces. A general disadvantage of RL is that the progress of learning can be very slow and non-monotonic. However, by incorporating (partial) model knowledge, learning can be sped up [9].

In this paper we address two important issues: First, we propose a learning control structure within the PH

Email addresses: osprangers@gmail.com (O.R. Sprangers), g.a.delgadolopes@tudelft.nl (G.A.D. Lopes), r.babuska@tudelft.nl (R. Babuška).

framework that retains important properties of the PBC. To this end, first a parameterization of a particular type of PBC, called energy-balancing passivity-based control (EB-PBC), is proposed such that the PDE arising in EB-PBC can be split into a non-assignable part satisfying a matching condition following from the EB-PBC framework and an assignable part that can be parameterized. Then, by applying actor-critic reinforcement learning the parameterized part can be learned while automatically verifying the matching PDE. This can be seen as a paradigm shift from the traditional model-based control synthesis for PH systems: we do not seek to synthesize a controller in closed-form, but we aim instead to learn one with proper structural constraints. This brings a number of advantages: I) It allows to specify the control goal in a “local” fashion through a reward function, without having to consider the entire global behavior of the system. The simplest example to illustrate this idea is by considering a reward function to be 1 when the system is in a small neighborhood of the desired goal and 0 everywhere else [21]. The learning algorithm will eventually find a global control policy. In the model-based PBC synthesis counterpart one needs to specify a desired global Hamiltonian. II) Learning brings performance in addition to the intrinsic stability properties of PBC. The structure of RL is such that the rewards are maximized, and these can include performance criteria, such as minimal time, energy consumption, etc. III) Learning offers additional robustness and adaptability since it tolerates model uncertainty in the PH framework.

From a learning control point of view, we present a systematic way of incorporating a priori knowledge into the RL problem. The approach proposed in this paper yields, after learning, a controller that can be interpreted in terms of energy shaping control strategies. The same interpretability is typically not found in the traditional RL solutions.

Thus, this work combines the advantages of both aforementioned control techniques, PBC and RL, and mitigates some of their respective disadvantages. Historically, the trends in control synthesis have oscillated between performance and stabilization. PBC of PH systems is rooted in the stability of multi-domain nonlinear systems. By including learning we aim to address performance in the PH framework. In the experimental section of the paper, we show that our method is also robust to unmodeled nonlinearities, such as control input saturation. Control input saturation in PBC for PH systems has been addressed explicitly in the literature [1,3,6,8,13,14,20]. We show that our approach solves the problem of control input saturation on the learning side without the need of augmenting the model-based PBC.

The work presented in this paper draws an interesting parallel with the application of iterative feedback tuning (IFT) [10] in the PH framework [7]. Both techniques optimize the parameters of the controller on-

line, with the difference that in IFT the objective is to minimize the error between the desired output and the measured output of the system, while our approach aims at maximizing a reward function, that can be very general. The choice of RL is warranted by its semi-supervised paradigm, as opposed to other traditional fully-supervised learning techniques, such as artificial neural networks or fuzzy approximators where the control specification (function approximation information) is input/output data instead of reward functions. Such fully-supervised techniques can be used within RL as function approximators to represent the value functions and control policies. Genetic algorithms can also be considered as an alternative to RL since they rely on fitness functions that are analogous to the rewards functions of RL. We aim to explore such classes of algorithms in our future work.

The theoretical background on PH systems and actor-critic reinforcement learning is described in Section 2 and Section 3, respectively. In Section 4, our proposal for a parameterization of input-saturated EB-PBC control, compatible with actor-critic reinforcement learning, is introduced. We then specialize this result to mechanical systems in Section 5. Section 6 provides simulation and experimental results for the problem of swinging up an input-saturated inverted pendulum and Section 7 concludes the paper.

2 Port-Hamiltonian Systems

Port-Hamiltonian (PH) systems are a natural way of representing a physical system in terms of its energy exchange with the environment through ports [18]. The general framework of PH systems was introduced in [15] and was formalized in [24,23]. In this paper, we consider the input-state-output representation of the PH system which is of the form¹:

$$\Sigma : \begin{cases} \dot{x} = [J(x) - R(x)] \nabla_x H(x) + g(x)u \\ y = g^T(x) \nabla_x H(x) \end{cases} \quad (1)$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$, $m \leq n$ is the control input, $J, R : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ with $J(x) = -J(x)^T$ and $R(x) = R(x)^T \geq 0$ are the interconnection and damping matrix, respectively, $H : \mathbb{R}^n \rightarrow \mathbb{R}$ the Hamiltonian which is the stored energy in the system, $u, y \in \mathbb{R}^m$ are conjugated variables whose product has the unit of power and $g : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ is the input matrix assumed to be full rank. For the remainder of this paper, we denote:

$$F(x) := J(x) - R(x) \quad (2)$$

¹ We use the notation $\nabla_x := \partial/\partial x$. Furthermore, all (gradient) vectors are column vectors.

This matrix satisfies $F(x) + F^T(x) = -2R(x) \leq 0$. System (1) satisfies the power-balance equation [17]:

$$\begin{aligned}\dot{H}(x) &= (\nabla_x H(x))^T \dot{x} \\ &= -(\nabla_x H(x))^T R(x) \nabla_x H(x) + u^T y\end{aligned}\quad (3)$$

Since $R(x) \geq 0$, we obtain:

$$\dot{H}(x) \leq u^T y \quad (4)$$

which is called the passivity inequality, if $H(x)$ is positive semi-definite, and cyclo-passivity inequality, if $H(x)$ is not positive semi-definite nor bounded from below [17]. Hence, systems satisfying (4) are called (cyclo-)passive systems. The goal is to obtain the target closed-loop system:

$$\Sigma_{cl} : \dot{x} = [J(x) - R_d(x)] \nabla_x H_d(x) \quad (5)$$

through *energy shaping* using EB-PBC [17] and *damping injection*, such that $H_d(x)$ is the desired closed-loop energy which has a minimum at the desired equilibrium x^* and satisfies:

$$\dot{H}_d(x) = -(\nabla_x H_d(x))^T R_d(x) \nabla_x H_d(x) \quad (6)$$

which implies (cyclo-)passivity according to (3)-(4) if the desired damping $R_d(x) \geq 0$. Hence, the control objective is achieved by rendering the closed-loop system passive with respect to the desired storage function $H_d(x)$.

2.1 Energy Shaping

Define the added energy function:

$$H_a(x) := H_d(x) - H(x) \quad (7)$$

A state-feedback law $u_{es}(x)$ is said to satisfy the energy-balancing property if it satisfies:

$$\dot{H}_a(x) = -u_{es}^T(x) y \quad (8)$$

If (8) holds, the desired energy $H_d(x)$ is the difference between the stored and supplied energy. Assuming $g(x) \in \mathbb{R}^{n \times m}$, $m < n$, $\text{rank}\{g(x)\} = m$, the control law:

$$u_{es}(x) = g^\dagger(x) F(x) \nabla_x H_a(x) \quad (9)$$

with $g^\dagger(x) = (g^T(x)g(x))^{-1}g^T(x)$ solves the EB-PBC problem with $H_a(x)$ a solution of the following set of PDE's [17]:

$$\begin{bmatrix} g^\perp(x) F^T(x) \\ g^T(x) \end{bmatrix} \nabla_x H_a(x) = 0 \quad (10)$$

with $g^\perp(x) \in \mathbb{R}^{(n-m) \times n}$ the full rank left-annihilator of $g(x)$, i.e. $g^\perp(x)g(x) = 0$.

2.2 Damping Injection

Damping is injected by feeding back the (new) passive output $g^T(x) \nabla_x H_d(x)$,

$$u_{di}(x) = -K(x)g^T(x) \nabla_x H_d(x) \quad (11)$$

with $K(x) \in \mathbb{R}^{m \times m}$, $K(x) = K^T(x) \geq 0$ such that:

$$R_d(x) = R(x) + g(x)K(x)g^T(x) \quad (12)$$

Hence, the full control law consists of an energy shaping part and a damping injection part:

$$\begin{aligned}u(x) &= u_{es}(x) + u_{di}(x) \\ &= g^\dagger(x)F(x)\nabla_x H_a(x) \\ &\quad - K(x)g^T(x)\nabla_x H_d(x)\end{aligned}\quad (13)$$

3 Actor-Critic Reinforcement Learning

In reinforcement learning, the system to be controlled (called ‘environment’ in the RL literature) is modeled as a Markov decision process (MDP). In a deterministic setting, this MDP is defined by the tuple $M(X, U, f, \rho)$, where X is the state space, U the action space and $f : X \times U \rightarrow X$ the state transition function that describes the process to be controlled that returns the state x_{k+1} after applying action u_k in state x_k . The vector x_k is obtained by applying a zero-order hold discretization $x_k = x(kT_s)$ with T_s the sampling time. The reward function is defined by $\rho : X \times U \rightarrow \mathbb{R}$ and returns a scalar reward $r_{k+1} = \rho(x_{k+1}, u_k)$ after each transition. The goal of RL is to find an optimal control policy $\pi : X \rightarrow U$ by maximizing an expected cumulative or total reward described as some function of the immediate expected rewards. In this paper, we consider a discounted sum of rewards. The value function $V^\pi : X \rightarrow \mathbb{R}$,

$$\begin{aligned}V^\pi(x) &= \sum_{i=0}^{\infty} \gamma^i r_{k+i+1}^\pi \\ &= \sum_{i=0}^{\infty} \gamma^i \rho(x_{k+i+1}, \pi(x_{k+i})), \quad x = x_k\end{aligned}\quad (14)$$

approximates this discounted sum during learning while following policy π where $\gamma \in [0, 1)$ is the discount factor.

When dealing with large and/or continuous state and action spaces, it is necessary to approximate the value function and policy. Actor-critic (AC) algorithms [4,11] learn a separate actor (policy π) and critic (value function V^π). The critic approximates and updates (improves) the value function. Then, the actor’s parameters are updated in the direction of that improvement. The actor

and critic are usually defined by a differentiable parameterization such that gradient ascent can be used to update the parameters. This is beneficial when dealing with continuous action spaces [22]. In this paper, the temporal-difference based Standard Actor-Critic (S-AC) algorithm from [9] is used. Define the approximated policy $\hat{\pi} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and the approximated value function as $\hat{V} : \mathbb{R}^n \rightarrow \mathbb{R}$. Denote the parameterization of the actor by $\vartheta \in \mathbb{R}^p$ and of the critic by $\theta \in \mathbb{R}^q$. The temporal difference [21]:

$$\delta_{k+1} := r_{k+1} + \gamma \hat{V}(x_{k+1}, \theta_k) - \hat{V}(x_k, \theta_k) \quad (15)$$

is used to update the critic parameters using the following gradient ascent update rule:

$$\theta_{k+1} = \theta_k + \alpha_c \delta_{k+1} \nabla_{\theta} \hat{V}(x_k, \theta_k) \quad (16)$$

in which $\alpha_c > 0$ is the learning rate. Eligibility traces $e_k \in \mathbb{R}^q$ [21] can be used to speed up learning by including reward information about previously visited states. The update for the critic parameters becomes:

$$e_{k+1} = \gamma \lambda e_k + \nabla_{\theta} \hat{V}(x_k, \theta_k) \quad (17)$$

$$\theta_{k+1} = \theta_k + \alpha_c \delta_{k+1} e_{k+1} \quad (18)$$

with $\lambda \in [0, 1)$ the trace-decay rate. The policy approximation can be updated in a similar fashion, as described below. RL needs exploration in order to visit new, unseen parts of the state-action space so as to possibly find better policies. This is achieved by perturbing the policy with a exploration term Δu_k . Many techniques have been developed for choosing the type of exploration term (see e.g. [2]). In this paper we consider Δu_k to be random with zero-mean. In the experimental section we choose Δu_k to be have a normal distribution. The overall control action now becomes:

$$u_k = \hat{\pi}(x_k, \vartheta_k) + \Delta u_k \quad (19)$$

The policy update is such that the policy parameters are updated towards (away from) Δu_k if the temporal difference (15) is positive (negative). This leads to the following policy update rule:

$$\vartheta_{k+1} = \vartheta_k + \alpha_a \delta_{k+1} \Delta u_k \nabla_{\vartheta} \hat{\pi}(x_k, \vartheta_k) \quad (20)$$

with $\alpha_a > 0$ the actor learning rate.

4 Energy-Balancing Actor-Critic

In this section we present our main results. Our approach is that we will use the PDE (10) and split it into an assignable, parameterizable part and an unassignable part that satisfies the matching condition. In this way, it is possible to parameterize the desired closed-loop

Hamiltonian $H_d(x)$ and simultaneously satisfy (10). After that, we parameterize the damping matrix $K(x)$. The two parameterized variables — the desired closed-loop energy $H_d(x)$ and damping $K(x)$ — are then suitable for Actor-Critic RL by defining two actors for these variables. First, we reformulate the PDE (10) in terms of the desired closed-loop energy $H_d(x)$ by applying (7):

$$\underbrace{\begin{bmatrix} g^{\perp}(x) F^T(x) \\ g^T(x) \end{bmatrix}}_{A(x)} (\nabla_x H_d(x) - \nabla_x H(x)) = 0 \quad (21)$$

and we denote the kernel of $A(x)$ as:

$$\ker(A(x)) = \{N(x) \in \mathbb{R}^{n \times b} : A(x)N(x) = 0\} \quad (22)$$

such that (21) reduces to:

$$\nabla_x H_d(x) - \nabla_x H(x) = N(x)a \quad (23)$$

with $a \in \mathbb{R}^b$. Suppose that (an example is given further on) the state vector x can be split, such that $x = [w^T \ z^T]^T$, where $z \in \mathbb{R}^c$ and $w \in \mathbb{R}^d$, $c + d = n$ corresponding to the zero and non-zero elements of $N(x)$ such that:

$$\begin{bmatrix} \nabla_w H_d(x) \\ \nabla_z H_d(x) \end{bmatrix} - \begin{bmatrix} \nabla_w H(x) \\ \nabla_z H(x) \end{bmatrix} = \begin{bmatrix} N_w(x) \\ 0 \end{bmatrix} a \quad (24)$$

We assume that the matrix $N_w(x)$ is rank d , which is always true for fully actuated mechanical systems (see Section 5). It is clear that $\nabla_z H_d(x) = \nabla_z H(x)$, which we call the matching condition, and hence $\nabla_z H_d(x)$ cannot be chosen freely. Thus, only the desired closed-loop energy gradient vector $\nabla_w H_d(x)$ is free for assignment. We consider a ξ -parameterized total desired energy with the following form:

$$\hat{H}_d(x, \xi) := H(x) + \xi^T \phi_H(w) + \bar{H}_d(w) + C \quad (25)$$

where $\xi^T \phi_H(w)$ represents a linear-in-parameters basis function approximator ($\xi \in \mathbb{R}^e$ a parameter vector and $\phi_H(w) \in \mathbb{R}^e$ the basis function, with e chosen sufficiently large to represent the assignable desired closed-loop energy), $\bar{H}_d(w)$ is an arbitrary function of w , and C chosen to render $\hat{H}_d(x, \xi)$ non-negative. The function $\hat{H}_d(x, \xi)$ automatically verifies (24). The elements of the desired damping matrix $K(x)$ of (13), denoted $\hat{K}(x, \Psi)$, can be parameterized in a similar way:

$$[\hat{K}(x, \Psi)]_{ij} = \sum_{l=1}^f [\Psi]_{ijl} [\phi_K(x)]_l \quad (26)$$

with $\Psi \in \mathbb{R}^{m \times m \times f}$ and

$$[\Psi]_{ijl} = [\Psi]_{jil} \quad (27)$$

a parameter vector such that $\hat{K}(x, \Psi) = \hat{K}^T(x, \Psi)$, $(i, j) = 1, \dots, m$ and $\phi_K(x) \in \mathbb{R}^f$ basis functions. We purposefully do not impose $\hat{K}(x, \Psi) \geq 0$ to allow the injection of energy in the system via the damping term. Although this breaches the passivity criterion of (4) we shall see that local stability can still be numerically demonstrated using passivity analysis in Section 6.3. This choice is made based on the knowledge that the standard Energy Balancing PBC method (without any extra machinery to accommodate saturation) cannot stabilize in the up position a saturated-input pendulum system starting from the down position. As such, this choice illustrates the power of RL in finding alternative routes to obtain control policies, such as injecting energy through the damping term. In other settings enforcing that $\hat{K}(x, \Psi) \geq 0$ benefits the stability analysis. The control law (13) now becomes (when no ambiguity is present, the function arguments are dropped to improve readability):

$$\begin{aligned} u(x, \xi, \Psi) &= g^\dagger(x) F(x) \begin{bmatrix} \nabla_w \hat{H}_d(x, \xi) - \nabla_w H(x) \\ 0 \end{bmatrix} \\ &\quad - \hat{K}(x, \Psi) g^T(x) \nabla_x \hat{H}_d(x, \xi) \\ &= g^\dagger F \begin{bmatrix} (\xi^T \nabla_w \phi_H)^T + \nabla_w \bar{H}_d \\ 0 \end{bmatrix} \\ &\quad - \hat{K} g^T \begin{bmatrix} (\xi^T \nabla_w \phi_H)^T + \nabla_w \bar{H}_d + \nabla_w H \\ \nabla_z H \end{bmatrix} \end{aligned} \quad (28)$$

Now, we are ready to introduce the update equations for the parameter vectors ξ , $[\Psi]_{ij}$. Denote by ξ_k , $[\Psi_k]_{ij}$ the value of the parameters at the discrete time step k . The policy $\hat{\pi}$ of the actor-critic reinforcement learning algorithm is chosen equal to the control law parameterized by (28):

$$\hat{\pi}(x_k, \xi_k, \Psi_k) := u(x_k, \xi_k, \Psi_k) \quad (29)$$

In this paper we take the control input saturation problem into account by considering a generic saturation function $\varsigma : \mathbb{R}^m \rightarrow \bar{S}$, $\bar{S} \subset \mathbb{R}^m$, such that:

$$\varsigma(u(x)) \in \bar{S} \quad \forall u \quad (30)$$

where \bar{S} is the set of valid control inputs. The control action with exploration (19) becomes:

$$u_k = \varsigma(\hat{\pi}(x_k, \xi_k, \psi_k) + \Delta u_k) \quad (31)$$

where Δu_k is drawn from a desired distribution. The exploration term to be used in the actor update (20)

must be adjusted to respect the saturation:

$$\Delta \bar{u}_k = u_k - \hat{\pi}(x_k, \xi_k, \psi_k) \quad (32)$$

Furthermore, we obtain the following gradients of the saturated policy:

$$\nabla_\xi \varsigma(\hat{\pi}) = \nabla_{\hat{\pi}} \varsigma(\hat{\pi}) \nabla_\xi \hat{\pi} \quad (33)$$

$$\nabla_{[\Psi]_{ij}} \varsigma(\hat{\pi}) = \nabla_{\hat{\pi}} \varsigma(\hat{\pi}) \nabla_{[\Psi]_{ij}} \hat{\pi} \quad (34)$$

Although not explicitly indicated in the previous equations, the (lack of) differentiability of the saturation function ς has to be considered for the problem at hand such that the computation of the gradient of ς can be made. For a traditional saturation in $u_i \in \mathbb{R}$ of the form $\max(u_{\min}, \min(u_{\max}, u_i))$, i.e. assuming each input u_i is bounded by u_{\min} and u_{\max} , then the gradient of ς is the zero matrix outside the unsaturated set S (i.e. when $u_i < u_{\min}$ or $u_i > u_{\max}$). For other types of saturation the function $\nabla \varsigma$ must be computed. Finally, the actor parameters ξ_k , $[\Psi_k]_{ij}$ are updated according to (20), respecting the saturated policy gradients. For the parameters of the desired Hamiltonian we obtain:

$$\xi_{k+1} = \xi_k + \alpha_{a,\xi} \delta_{k+1} \Delta \bar{u}_k \nabla_\xi \varsigma(\hat{\pi}(x_k, \xi_k, \Psi_k)) \quad (35)$$

and for the parameters of the desired damping we have:

$$\begin{aligned} [\Psi_{k+1}]_{ij} &= [\Psi_k]_{ij} + \\ &\quad \alpha_{a,[\Psi]_{ij}} \delta_{k+1} \Delta \bar{u}_k \nabla_{[\Psi_k]_{ij}} \varsigma(\hat{\pi}(x_k, \xi_k, \Psi_k)) \end{aligned} \quad (36)$$

where $(i, j) = 1, \dots, m$, while observing (27). Algorithm 1 gives the entire Energy-Balancing Actor-Critic algorithm with input saturation.

The dynamics of the Energy-Balancing Actor Critic Algorithm 1 raises a number of questions regarding stability and convergence: are the good stability properties of the traditional energy-balancing PBC lost? In effect this is not the case, as if the parameter ξ is fixed then stability is preserved, in the sense that the system is passive to the storage function $\hat{H}_d(x, \xi)$. The question is then if during learning (while the parameter ξ is evolving) will the Hamiltonian $\hat{H}_d(x, \xi)$ capture the desired control specification. One cannot assume that the desired Hamiltonian will immediately fulfill the control specification, since if that was the case then no learning is needed. In the RL community it is generally accepted that during learning no stability and convergence guarantees can be given [21], as exploration is a necessary component of the framework. In our framework, we cannot guarantee convergence during learning, but by constraining the desired Hamiltonian we can prevent the total energy to grow unbounded, avoiding possible instabilities. Another relevant question is: will RL converge in this setting? The constrained Hamiltonian results in sub-optimal policies from the RL side as the representation of the policy is not entirely free. In practice, as

Algorithm 1 Energy-Balancing Actor-Critic

Input: System (1), λ , γ , α_a for each actor, α_c .

```
1:  $e_0(x) = 0 \quad \forall x$ 
2: Initialize  $x_0, \theta_0, \xi_0, \Psi_0$ 
3:  $k \leftarrow 1$ 
4: loop
5:   Execute:
6:     Draw  $\Delta u_k \sim \mathcal{N}(0, \sigma^2)$ , calculate action  $u_k = \varsigma(\hat{\pi}(x_k, \xi_k, \psi_k) + \Delta u_k)$ ,  $\Delta \bar{u}_k = u_k - \hat{\pi}(x_k, \xi_k, \psi_k)$ 
7:     Observe next state  $x_{k+1}$  and calculate reward  $r_{k+1} = \rho(x_{k+1}, u_k)$ 
8:   Critic:
9:     Temporal difference:  $\delta_{k+1} = r_{k+1} + \gamma \hat{V}(x_{k+1}, \theta_k) - \hat{V}(x_k, \theta_k)$ 
10:    Eligibility trace:  $e_{k+1} = \gamma \lambda e_k + \nabla_{\theta} \hat{V}(x_k, \theta_k)$ 
11:    Critic update:  $\theta_{k+1} = \theta_k + \alpha_c \delta_{k+1} e_{k+1}$ 
12:   Actors:
13:     Actor 1 ( $\hat{H}_d(x, \xi)$ ):  $\xi_k + \alpha_{a,\xi} \delta_{k+1} \Delta \bar{u}_k \nabla_{\xi} \varsigma(\hat{\pi}(x_k, \xi_k, \Psi_k))$ 
14:     Actor 2 ( $\hat{K}(x, \Psi)$ ):
15:       for  $i, j = 1, \dots, m$  do
16:          $[\Psi_{k+1}]_{ij} = [\Psi_k]_{ij} + \alpha_{a, [\Psi]_{ij}} \delta_{k+1} \Delta \bar{u}_k \nabla_{[\Psi_k]_{ij}} \varsigma(\hat{\pi}(x_k, \xi_k, \Psi_k))$ 
17:       end for
18: end loop
```

presented next in Section 6, we observe that not only will the RL component converge very fast (faster than traditional model-free RL) but throughout learning the system never gets unstable. Additionally, the resulting policy performs as well as standard model-free approximated RL algorithms. The last point to consider is that since the Energy Balancing PBC suffers from the dissipation obstacle [17], limiting its applicability to special classes of systems such as mechanical systems, the algorithm we present contains the same limitation. Eliminating such limitation is open research for future work. “hello”

5 Mechanical Systems

To illustrate an application of the method, consider a fully actuated mechanical system of the form:

$$\Sigma_m : \begin{cases} \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -I & -\bar{R} \end{bmatrix} \begin{bmatrix} \nabla_q H(q, p) \\ \nabla_p H(q, p) \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u \\ y = \begin{bmatrix} 0 & G \end{bmatrix} \begin{bmatrix} \nabla_q H(q, p) \\ \nabla_p H(q, p) \end{bmatrix} \end{cases} \quad (37)$$

with $q \in \mathbb{R}^{\bar{n}}$, $p \in \mathbb{R}^{\bar{n}}$ ($\bar{n} = \frac{n}{2}$, n even) the generalized positions and momenta, respectively, $G = I$ and $\bar{R} \in \mathbb{R}^{\bar{n} \times \bar{n}}$ the damping matrix. The system admits (1) with $\bar{R} > 0$ and the Hamiltonian:

$$H(q, p) = \frac{1}{2} p^T M^{-1}(q) p + P(q) \quad (38)$$

with $M(q) = M^T(q) > 0$ the inertia matrix and $P(q)$ the potential energy. For the system (37) it holds that

$\text{rank } \{g(x)\} = \bar{n}$ and the state vector can be split into part $w = [q_1, q_2, \dots, q_{\bar{n}}]^T$ and part $z = [p_1, p_2, \dots, p_{\bar{n}}]^T$. Since $g(x) = [0 \ I]^T$ its annihilator can be written as $g^\perp(x) = [\bar{g}(x) \ 0]$, for an arbitrary matrix $\bar{g}(x)$. This means that only the potential energy can be shaped, which is widely known in EB-PBC for mechanical systems. The approximated desired closed-loop energy (25) reads:

$$\hat{H}_d(x, \xi) = \frac{1}{2} p^T M^{-1}(q) p + \xi^T \phi_H(q) \quad (39)$$

where the first term represents the unassignable part, i.e. the kinetic energy of the system Hamiltonian (38), and the second term $\xi^T \phi_H(q)$ the assignable desired potential energy. The actor updates can be defined for each parameter according to (35)–(36). For underactuated mechanical systems, e.g. $G = [0 \ I]^T$, the split state vector z is enlarged with those q -coordinates that cannot be actuated directly because these coordinates correspond to the zero elements of $N(x)$, (i.e. the matrix $N_q(x)$ is no longer rank \bar{n}).

6 Example: Pendulum Swing-up

To validate our method, the problem of swinging up an inverted pendulum subject to control saturation is studied in simulation and using the actual physical setup depicted in Fig. 1.

The pendulum swing-up is a low-dimensional, but highly nonlinear control problem commonly used as a benchmark in the RL literature [9] and it has also been studied in PBC [3]. The equations of motion admit (37) and

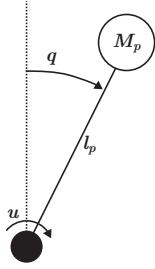


Fig. 1. Inverted pendulum setup.

read:

$$\Sigma_p : \begin{cases} \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -\bar{R}(\dot{q}) \end{bmatrix} \begin{bmatrix} \nabla_q H(q, p) \\ \nabla_p H(q, p) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_p}{R_p} \end{bmatrix} u \\ y = \begin{bmatrix} 0 & \frac{K_p}{R_p} \end{bmatrix} \begin{bmatrix} \nabla_q H(q, p) \\ \nabla_p H(q, p) \end{bmatrix} \end{cases} \quad (40)$$

with q the angle of the pendulum and p the angular momentum, thus we denote the full measurable state $x = [q, p]^T$. The damping term is:

$$\bar{R}(\dot{q}) = b_p + \frac{K_p^2}{R_p} + \frac{\sigma_p}{|\dot{q}|} \quad (41)$$

for which it holds that $\bar{R}(\dot{q}) > 0, \forall \dot{q}$. Furthermore, we denote the Hamiltonian:

$$H(q, p) = \frac{p^2}{2J_p} + P(q) \quad (42)$$

with:

$$P(q) = M_p g_p l_p (1 + \cos q) \quad (43)$$

The model parameters are given in Table 1. The desired Table 1

Inverted pendulum model parameters

Model parameters	Symbol	Value	Units
Pendulum inertia	J_p	$1.90 \cdot 10^{-4}$	kgm^2
Pendulum mass	M_p	$5.2 \cdot 10^{-2}$	kg
Gravity	g_p	9.81	m/s^2
Pendulum length	l_p	$4.20 \cdot 10^{-2}$	m
Dynamic friction	b_p	$2.48 \cdot 10^{-6}$	Nms
Static friction	σ_p	$1.0 \cdot 10^{-3}$	N
Torque constant	K_p	$5.60 \cdot 10^{-2}$	Nm/A
Rotor resistance	R_p	9.92	Ω

Hamiltonian (25) reads:

$$\hat{H}_d(x, \xi) = \frac{p^2}{2J_p} + \xi^T \phi_H(q) \quad (44)$$

Only the potential energy can be shaped that we denote by $\hat{P}_d(q, \xi) = \xi^T \phi_H(q)$. Furthermore, as there is only one input, $\hat{K}(x, \Psi)$ becomes a scalar:

$$\hat{K}(x, \psi) = \psi^T \phi_K(x) \quad (45)$$

Thus, control law (28) results in:

$$u(x, \xi, \psi) = g^\dagger F \begin{bmatrix} \xi^T \nabla_q \phi_H - \nabla_q P \\ 0 \end{bmatrix} - \hat{K} g^T \begin{bmatrix} \xi^T \nabla_q \phi_H \\ J_p^{-1} p \end{bmatrix} \\ = -\frac{R_p}{K_p} (\xi^T \nabla_q \phi_H + M_p g_p l_p \sin(q)) \quad (46)$$

$$- \frac{K_p}{R_p} \psi^T \phi_K \dot{q} \quad (47)$$

which we define as the policy $\hat{\pi}(x, \xi, \psi)$. Hence, we have two actor updates:

$$\xi_{k+1} = \xi_k + \alpha_{a,\xi} \delta_{k+1} \Delta \bar{u}_k \nabla_{\xi} \varsigma(\hat{\pi}(x_k, \xi_k, \psi_k)) \quad (48)$$

$$\psi_{k+1} = \psi_k + \alpha_{a,\psi} \delta_{k+1} \Delta \bar{u}_k \nabla_{\psi} \varsigma(\hat{\pi}(x_k, \xi_k, \psi_k)) \quad (49)$$

for the desired potential energy $\hat{P}_d(q, \xi)$ and the desired damping $\hat{K}(x, \psi)$, respectively.

6.1 Function Approximation

To approximate the critic and the two actors, function approximators are necessary. In this paper we use the Fourier basis [12] because of its ease of use, the possibility to incorporate information about the symmetry in the system and the ability to ascertain properties useful for stability analysis of this specific problem. The periodicity of the function approximators obtained via a Fourier basis is compatible with the topology of the configuration space of the pendulum, defined to be $S^1 \times \mathbb{R}$. We define a multivariate N th-order² Fourier basis for n dimensions as:

$$\phi(\bar{x}) = \sum_{i=1}^{(N+1)^n} \cos(\pi c_i^T \bar{x}) \quad (50)$$

with $c_i \in \mathbb{Z}^n$, which means that all possible $N+1$ integer values, or frequencies, are combined in a vector in \mathbb{Z}^n to create a matrix $c \in \mathbb{Z}^{n \times (N+1)^n}$ containing all possible frequency combinations. For example,

$$c_1 = [0 \ 0]^T, \ c_2 = [1 \ 0]^T, \ \dots, \ c_{(3+1)^2} = [4 \ 4]^T \quad (51)$$

² ‘Order’ refers to the order of approximation; ‘dimensions’ to the number of states in the system.

for a 3rd-order Fourier basis in 2 dimensions. The state x is scaled according to:

$$\bar{x}_i = \frac{x_i - x_{i,\min}}{x_{i,\max} - x_{i,\min}} (\bar{x}_{i,\max} - \bar{x}_{i,\min}) + \bar{x}_{i,\min} \quad (52)$$

for $i = 1, \dots, n$ with $(\bar{x}_{i,\min}, \bar{x}_{i,\max}) = (-1, 1)$. Projecting the state variables onto this symmetrical range has several advantages. First, this means that the policy will be periodic with period $T = 2$, such that it wraps around (i.e., modulo 2π) and prevents discontinuities at the boundary values of the angle ($x = [\pi \pm \epsilon, p]$, ϵ very small). Second, learning will be faster because updating the value function and policy for some x also applies to the sign-opposite value of x . Third, $\dot{P}_d(0, \xi) = 0$ by the choice of parameterization, which is beneficial for stability analysis. Although the momentum will now also be periodic in the value function and policy, this is not a problem because the value function and policy approximation are restricted to a domain and the momentum itself is also restricted to the same domain using saturation. We adopt the adjusted learning rate from [12] such that:

$$\alpha_{a_i, \xi} = \frac{\alpha_{a_b, \xi}}{\|c_i\|_2}, \quad \alpha_{a_i, \psi} = \frac{\alpha_{a_b, \psi}}{\|c_i\|_2} \quad (53)$$

for $i = 1, \dots, (N+1)^n$ with $\alpha_{a_b, \xi}$, $\alpha_{a_b, \psi}$ the base learning rate for the two actors (Table 2) and $\alpha_{a_1, \xi} = \alpha_{a_b, \xi}$, $\alpha_{a_1, \psi} = \alpha_{a_b, \psi}$ to avoid division by zero for $c_1 = [0, 0]^T$. Equation (53) implies that parameters corresponding to basis functions with higher (lower) frequencies are learned slower (faster). The parameterizations described above result in $\hat{H}_d(x^*, \xi) = 0$ for all ξ , where $x^* = [0, 0]^T$ is the goal state. This entails that the goal state is a critical point of the Hamiltonian throughout the learning process, in effect speeding up the RL convergence.

6.2 Simulation

The task is to learn to swing up and stabilize the pendulum from the initial position pointing down $x_0 = [\pi, 0]^T$ to the desired equilibrium position at the top $x^* = [0, 0]^T$. Since the control action is saturated, the system is not able to swing up the pendulum directly, but rather it must swing back and forth to build up momentum to eventually reach the equilibrium. The reward function ρ is defined such that it has its maximum in the desired unstable equilibrium and penalizes other states via:

$$\rho(x, u) = Q_r (\cos(q) - 1) - R_r p^2 \quad (54)$$

with:

$$Q_r = 25, \quad R_r = \frac{0.1}{J_p^2} \quad (55)$$

This reward function is consistent with the mapping $S^1 \rightarrow \mathbb{R}$ for the angle and proved to improve performance over a purely quadratic reward, such as the one

used in e.g. [9]. For the critic, we define the basis function approximation as:

$$\hat{V}(x, \theta) = \theta^T \phi_c(x) \quad (56)$$

with $\phi_c(x)$ a 3rd-order Fourier basis resulting in 16 learnable parameters θ in the domain $[q_{\min}, q_{\max}] \times [p_{\min}, p_{\max}] = [-\pi, \pi] \times [-8\pi J_p, 8\pi J_p]$. Actor 1 ($\hat{P}_d(q, \xi)$) is parameterized using a 3rd-order Fourier basis in the range $[-\pi, \pi]$ resulting in 4 learnable parameters. Actor 2 ($\hat{K}(x, \psi)$) is also parameterized using a 3rd-order Fourier basis for the full state space, in the same domain as the critic. Exploration is done at every time step by randomly perturbing the action with a normally distributed zero-mean white noise with standard deviation $\sigma = 1$, i.e.:

$$\Delta u \sim \mathcal{N}(0, 1) \quad (57)$$

We incorporate saturation by defining the saturation function (30) as:

$$\varsigma(u_k) = \begin{cases} u_k & \text{if } |u_k| \leq u_{\max} \\ \text{sgn}(u_k)u_{\max} & \text{otherwise} \end{cases} \quad (58)$$

Recall that the saturation must be taken into account in the policy gradients by applying (33)-(34). The parameters were all initialized with zero vectors of appropriate dimensions, i.e. $(\theta_0, \xi_0, \psi_0) = 0$. The algorithm was first run with the system simulated in Matlab for 200 trials of three seconds each (with a near-optimal policy, the pendulum needs approximately one second to swing up). Each trial begins in the initial position x_0 . This simulation was repeated 50 times to get an estimate of the average, minimum, maximum and confidence regions for the learning curve. The simulation parameters are given in Table 2.

Table 2
Simulation parameters

Simulation parameters	Symbol	Value	Units
Number of trials	—	200	-
Trial duration	T_t	3	s
Sample time	T_s	0.03	s
Decay rate	γ	0.97	-
Eligibility trace decay	λ	0.65	-
Exploration variance	σ^2	1	-
Max control input	u_{\max}	3	V
Learning rate of critic	α_c	0.05	-
Learning rate of $\hat{P}_d(q, \xi)$	$\alpha_{a_b, \xi}$	1×10^{-10}	-
Learning rate of $\hat{K}(x, \psi)$	$\alpha_{a_b, \psi}$	0.2	-

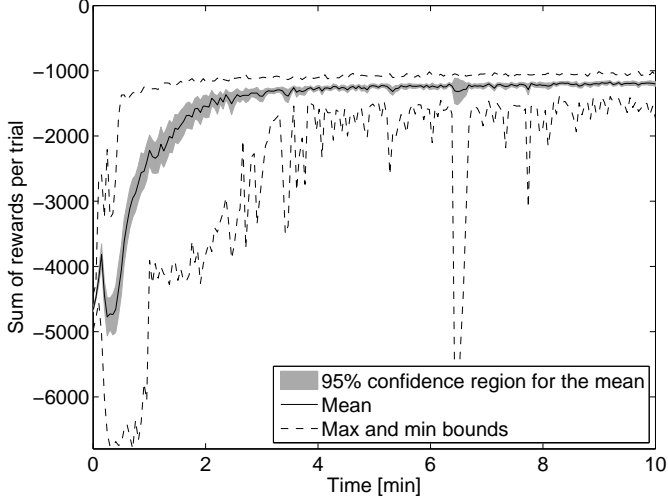


Fig. 2. Results for the EBAC method for 50 learning simulations.

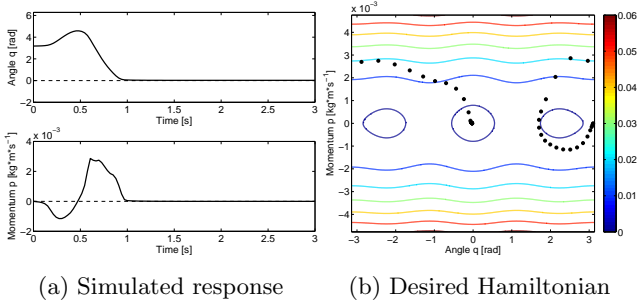


Fig. 3. Simulation results for the angle q (a, top), momentum p (a, bottom) and the desired closed-loop Hamiltonian $H_d(x, \xi, \psi)$ (b) including the simulated trajectory (black dots) using the policy learned.

Fig. 2 shows the average learning curve obtained after 50 simulations. The algorithm shows good convergence and on average needs about 2 minutes (40 trials) to reach a near-optimal policy. The initial drop in performance is caused by the zero-initialization of the value function (critic), which is too optimistic compared to the true value function. Therefore, the controller explores a large part of the state space and receives a lot of negative rewards before it learns the true value of the states. A simulation using the policy learned in a typical experiment is given in Fig. 3a. As can be seen, the pendulum swings back once to build up momentum to eventually get to the equilibrium. The desired Hamiltonian $\hat{H}_d(x, \xi)$ (44), acquired through learning, is given in Fig. 3b. There are three minima, of which one corresponds to the desired equilibrium. The other two equilibria are undesirable wells that come from the shaped potential energy $\hat{P}_d(q, \xi)$ (Fig. 4a). These minima are the result of the algorithm trying to swing up the pendulum in a single swing, which is not possible due to the saturation. Hence, a swing-up strategy is necessary to avoid staying

in these wells. The number of these undesirable wells is a function of the control saturation and of the number of basis functions chosen to approximate $\hat{P}_d(q, \xi)$. The learned damping $\hat{K}(x, \psi)$ (Fig. 4b) is positive (white) towards the equilibrium thus extracting energy from the system, while it is negative (gray) in the region of the initial state. The latter corresponds to pumping energy into the system, which is necessary to build up momentum for the swing-up and to escape the undesirable wells of $\hat{P}_d(q, \xi)$. A disadvantage is that control law (47), with the suggested basis functions, is always zero for the set $\Omega = \{x \mid x = (0 + j\pi, 0), j = 1, 2, \dots\}$ which implies that it is zero not only at the desired equilibrium, but also at the initial state x_0 . During learning this is not a problem because there is constant exploration, but after learning the system should not be initialized in exactly x_0 otherwise it will stay in this set. It can be overcome by initializing with a small perturbation ϵ around x_0 . In real-life systems it will also be less a problem because there is generally noise present on the sensors.

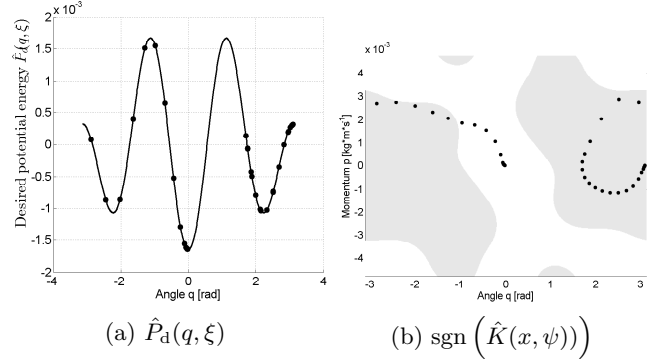


Fig. 4. Desired potential energy (a) and desired damping (b) (gray: negative; white: positive) for a typical learning experiment. The black dots indicate the value of the respective quantity for the simulation of Fig. 3a.

6.3 Stability of the Learned Controller

Since control saturation is present, the target dynamics do not satisfy (5). Hence, to conclude local stability of x^* based on (6), we calculate $\hat{H}_d(x, \xi)$ for the unsaturated case (Fig. 5a)³ and the saturated case ($\hat{H}_{d, \text{sat}}(x, \xi)$) and compute the sign of the difference (Fig. 5b). By looking at Fig. 5b, it appears that $\exists \delta \subset \mathbb{R}^n : |x - x^*| < \delta$ such that $\hat{H}_{d, \text{sat}}(x, \xi) = \hat{H}_d(x, \xi)$. It can be seen from Fig. 5b that such a δ exists, i.e., a small gray region around the equilibrium x^* exists. Hence, we can use $\hat{H}_d(x, \xi)$ around x^* and assess stability using (6). From Fig. 3b it follows

³ Fig. 5a is sign-opposite to Fig. 4b, which is logical, because the negative (positive) regions of $\hat{K}(x, \psi)$ correspond to negative (positive) damping which corresponds to a positive (negative) value of $\hat{H}_d(x, \xi)$ based on (6).

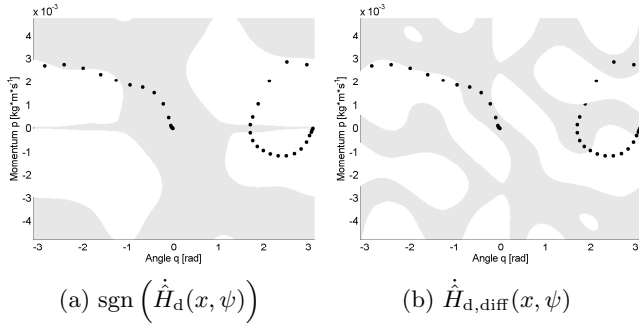


Fig. 5. Signum of $\dot{H}_d(x, \psi)$ (a) indicating positive (white) and negative (gray) regions and (b) $\dot{H}_{d,diff}(x, \psi) = \text{sgn}(\dot{H}_d(x, \psi) - \dot{H}_{d,sat}(x, \psi))$ indicating regions where $\dot{H}_d(x, \psi) = \dot{H}_{d,sat}(x, \psi)$ (gray) and $\dot{H}_d(x, \psi) \neq \dot{H}_{d,sat}(x, \psi)$ (white). Black dots indicate the simulated trajectory.

that $\dot{H}_d(x, \xi) > 0$ for all states in Fig. 3b. From Fig. 4a we infer that locally,

$$\arg \min \hat{P}_d(q, \xi) = x^*; \quad \dot{\hat{P}}_d(x^*, \xi) = 0; \quad \ddot{\hat{P}}_d(x^*, \xi) > 0 \quad (59)$$

the latter two of which naturally result from the basis function definition. Furthermore, from Fig. 4b it can be seen that around x^* , $\hat{K}(x, \psi) > 0$. Hence, in a region δ around x^* ,

$$\dot{H}_d(x, \xi) > 0; \quad \dot{H}_d(x, \xi) \leq 0; \quad \dot{H}_d(x^*, \xi) = 0 \quad (60)$$

which implies local asymptotic stability of x^* . Extensive simulations show that similar behaviour is always achieved.

6.4 Real-time Experiments

Using the physical setup shown in Fig. 1, 20 learning experiments were run using identical settings as in the simulations. The result is given in Fig. 6. The algorithm shows slightly slower convergence - about 3 minutes of learning (60 trials) to reach a near-optimal policy instead of 40 - and a less consistent average when compared to Fig. 2. This can be attributed to a combination of model mismatch and the symmetrical basis functions (through which it is not possible to incorporate non-symmetrical friction that is present in the real system). Overall though, the performance can be considered good when compared to the simulation results. Also, the same performance dip is present which can again be attributed to the optimistic value function initialization.

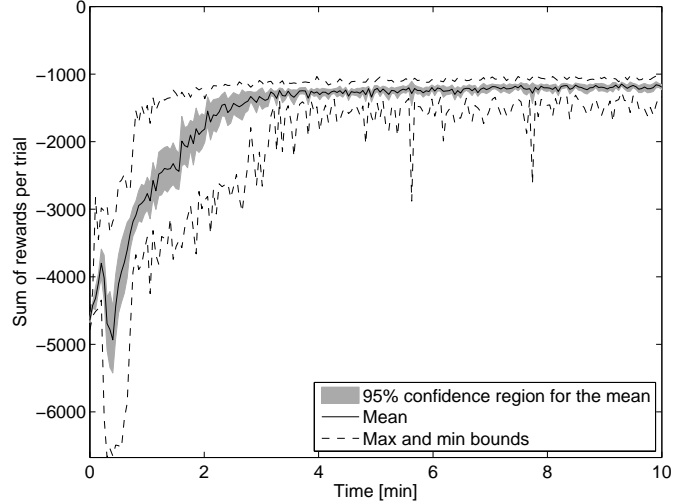


Fig. 6. Results for the EBAC method for 20 learning experiments with the real physical system.

7 Conclusions

In this paper, we have presented a method to systematically parameterize EB-PBC control laws that is robust to extra nonlinearities such as input control saturation. The parameters are then found by making use of actor-critic reinforcement learning. In this way, we are able to learn a closed-loop energy landscape for PH systems. The advantages are that optimal controllers can be generated using energy-based control techniques, there is no need to specify a global system Hamiltonian, and the solutions acquired by means of reinforcement learning can be interpreted in terms of energy shaping and damping injection, which makes it possible to numerically assess stability using passivity theory. By making use of the model knowledge the actor-critic method is able to quickly learn near-optimal policies. A drawback is that for multiple input systems, generating many actor updates for the desired damping matrix can be computationally expensive. We have found that the proposed Energy Balancing Actor Critic algorithm performs very well in a physical mechanical setup. Due to the intrinsic energy boundedness of the learned desired Hamiltonian, we have observed that the system never gets unstable during learning. We are currently active on the extension of the algorithms presented to an IDA-PBC setting, such that more classes of systems can be addressed and more freedom is given in shaping the desired Hamiltonian (e.g. Kinetic energy shaping).

References

- [1] Y. Wang A. Wei. Stabilization and H^∞ control of nonlinear port-controlled Hamiltonian systems subject to actuator saturation. *Automatica*, 46(12):2008–2013, 2010.
- [2] J. Asmuth, L. Li, M.L. Littman, A. Nouri, and D. Wingate. A Bayesian sampling approach to exploration in reinforcement

- learning. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 19–26. AUAI Press, 2009.
- [3] K.J. Åström, J. Aracil, and F. Gordillo. A family of smooth controllers for swinging up a pendulum. *Automatica*, 44:1841–1848, 2008.
- [4] A. Barto, R. Sutton, and C. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 13:835–846, 1983.
- [5] V. Duindam and S. Stramigioli. *Modeling and Control for Efficient Bipedal Walking Robots: A Port-Based Approach*. Springer, 2009.
- [6] G. Escobar, R. Ortega, and H. Sira-Ramirez. Output-feedback global stabilization of a nonlinear benchmark system using a saturated passivity-based controller. *IEEE Transactions on Control Systems Technology*, 7(2):289–293, 1999.
- [7] K. Fujimoto and I. Koyama. Iterative feedback tuning for Hamiltonian systems. In *Proceedings of the 17th IFAC World Congress*, pages 15678–15683, Seoul, Korea, 2008.
- [8] K. Fujimoto and T. Sugie. Iterative learning control of Hamiltonian systems: I/O based optimal control approach. *IEEE Transactions on Automatic Control*, 48(10):1756–1761, 2003.
- [9] I. Grondman, M. Vaandrager, L. Buşoniu, R. Babuška, and E. Schuitema. Efficient model learning methods for actor-critic control. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, In press.
- [10] H. Hjalmarsson. Iterative feedback tuning: an overview. *International Journal Adaptive Control and Signal Processing*, 16:373–395, 2002.
- [11] V.R. Konda and J.N. Tsitsiklis. On actor-critic algorithms. *SIAM Journal on Control and Optimization*, 42(4):1143–1166, 2003.
- [12] G. Konidaris and S. Osentoski. Value function approximation in reinforcement learning using the Fourier basis. Technical Report 101, Autonomous Learning Laboratory, Computer Science Department, University of Massachusetts Amherst, 2008.
- [13] A. Macchelli. *Port Hamiltonian Systems: A unified approach for modeling and control finite and infinite dimensional physical systems*. PhD thesis, University of Bologna, 2002.
- [14] A. Macchelli, C. Melchiorri, C. Secchi, and C. Fantuzzi. A variable structure approach to energy shaping. In *Proceedings of the European Control Conference*, 2003.
- [15] B. Maschke and A. van der Schaft. Port-controlled Hamiltonian systems: modelling origins and system theoretic properties. In *Proceedings of the third Conference on nonlinear control systems (NOLCOS)*, pages 282–288, 1992.
- [16] R. Ortega and M. W. Spong. Adaptive motion control of rigid robots: a tutorial. *Automatica*, 25:877–888, 1989.
- [17] R. Ortega, A. van der Schaft, F. Castanos, and A. Astolfi. Control by interconnection and standard passivity-based control of port-Hamiltonian systems. *IEEE Transactions on Automatic Control*, 53(11):2527–2542, 2008.
- [18] R. Ortega, A. van der Schaft, I. Mareels, and B. Maschke. Putting energy back in control. *IEEE Control Systems Magazine*, 21(2):18–33, 2001.
- [19] C. Secchi and S. Stramigioli. *Control of Interactive Robotic Interfaces*. Springer, 2007.
- [20] W. Sun, Z. Lin, and Y. Wang. Global asymptotic and finite-gain L_2 stabilization of port-controlled Hamiltonian systems subject to actuator saturation. In *Proceedings of the American Control Conference*, St. Louis, 2009.
- [21] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- [22] R. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12:1057–1063, 2000.
- [23] A. van der Schaft. *L_2 -Gain and Passivity Techniques in Nonlinear Control*. Springer, 2000.
- [24] A. van der Schaft and B. Maschke. On the Hamiltonian formulation of nonholonomic mechanical systems. *Reports on Mathematical Physics*, 34:225–233, 1994.